

CLIENT SIDE

Intelligence for Global Business

NEWS

TECH WRITER:

*Volumizing:
Good for Hair,
Not for Content*

TECH WRITER:

*Why Content
Authors Should
Care About TM*

LPM CORNER:

*Why
Technology
Can't Do the
Job*

JANUARY 2005 SUPPLEMENT

2004 Technical Writing Supplement

The Business Journal
for GILT Professionals
9.95 US
12.95 CAN
9.95 EURO



in this issue

2004 Tech Writing Supplement Volume 4 Issue S2

ClientSideNews

*“about the tech writing
focus supplement...”*

- 3 @issue—publisher's note
- 4 article 1—volumizing: good for hair, not for content
- 7 article 2—why content authors should care about TM
- 10 article 3—food for thought
- 13 article 4—why technology can't do the job
- 17 article 5—no Information like No-Information

intelligence for global business

ClientSide News Magazine is produced exclusively by:

ClientSide Publications
517 South Baldwin Street
Suite 100
Woodland Park, CO 80863
USA

Phone: 719.686-8759
FAX: 719-623.0394

Publisher
Shaun P. Daggett
sdaggett@clientsidenews.com

Advertising Sales
801.545.8816
adsales@clientsidenews.com

Send information and editorial submissions to
editor@clientsidenews.com

Send press releases to
news@clientsidenews.com

publication & membership

ClientSide News is published monthly by ClientSide Publications™, a division of ClientSide News LLC.

Annual subscriptions to ClientSide News Magazine are FREE to CSN Members. Contact CSN headquarters for membership information.

The mission of ClientSide News Magazine is to facilitate the exchange of information among professionals on the client side of the globalization, internationalization, localization and translation industry, to bring forward information and solutions, to serve as a forum for discussion of emerging trends and issues, to deliver intelligence for global business.

© 2004 ClientSide News LLC. All rights reserved. "CSN" and the CSN logo are service and trade marks registered in the United States and other nations.

Published articles and opinions do not necessarily reflect the views of the magazine or ClientSide News. ClientSide News is not responsible for loss, damage or other injury to manuscripts or other materials.

@issue

By Shaun P. Daggett, CSN Publisher

the tech writing focus supplement

In the summer of 2003, CSN and GALA had a one day session in Denver to discuss the needs of clients, and how service providers could meet those needs. CSN invited leading clients from its Chief Globalization Officer Board, and GALA invited the heads of leading technology companies and leading service providers to attend.

It was a great session with lots of meaningful dialog from both sides. In the end, no single localization solution, technology or service stood out as a leader. Each avenue explored came full circle back to content creation—the source. If you could reduce the number of words in the source language, clients would gain the time-to-market and cost savings they are so desperately seeking.

We did find one thing that gave clients immediate, tangible time and cost gains, and this was the one solution that came out of this meeting. Of course, CSN was interested in covering solutions that clients could tap into immediately. And ABREVE® met the need.

ABREVE® was developed by Hans Fenstermacher, Chairman of GALA and President and founder of ArchiText, a leading language service provider. It is a methodology for reducing volume and globalizing content that ArchiText has combined with its localization capability to create a patent-pending end-to-end localization process

This was exactly what we were looking for, and Hans's insights on this subject have been the source for each of the Tech Writing Focus articles CSN has published throughout 2004. This supplement is a compilation of all those articles. This has been no small feat for Hans, and through his efforts, our entire industry has benefited. CSN extends our thanks and gratitude for these great pieces of work—now he can add author to his long list of accomplishments in this industry.

volumizing:

*good for hair,
bad for content*

Life in the 21st century centers around information. In practically every waking moment, we create information, we receive it, we process it, we pass it on, we ignore it, but most of all, we need it. Those who process information for a living appear to have developed a relentless informational imperative: if it can be written, it must be written. Thus, many content developers fall into "volumizing" their content instead of preparing it for the global workflow and for end users.

Where does it all come from?

Technology has long had its own imperative, so products offer an abundance of choice, which we document equally abundantly. For example:

"The first step in creating a format from an existing file or folder is to create a new graphic FRM file by either clicking the New icon from the What To Do section of the Getting Started page and then clicking the Standard.frm icon from the Default template tab (as shown in the following image), or clicking New from the File menu, or clicking the down arrow on the New icon from the left side of the Standard toolbar and clicking Format."¹

Products also must do more and more, to ostensibly please a demanding public. Thus, features are continuously added. Naturally, these features are documented in abundance, or—and I trust the reader will note the irony here—how else would anyone know the features are even there?

Virtually no part of our lives in the Information Age is unscathed by the time squeeze, least of all content development efforts. Blaise Pascal once wrote (in French), "This letter is longer only because I did not have the leisure to make it shorter."

Precisely. We no longer have the leisure to make content efficient and shorter. We settle for quantity over quality. Witness the demise of editing at nearly every level of content development and localization.



By
Hans
Fenstermacher

Volume manifests itself in both obvious and devilishly sneaky ways. Take, for example, that bugbear of technical writing, the template. This is one of the more insidious ways content becomes "volumized." A template by definition calls for rigid adherence to principles—which are usually created in a vacuum (or at least a classroom). To wit:

"Minimum System Requirements

Make sure your system meets the following minimum requirements:"

The template here calls for a "bridging" sentence between the header and the bulleted list below. Good order (a.k.a. tradition) is preserved, but if you smell a rat, you've got a better nose than most technical writers. All this template does is make the user read the same thing twice—and increase the cost to create and localize the content (more about that later).

Then there are the San Andreas-sized structural faults that run through most content. They may take the form of chunky block text or text that describes a graphic in every detail, or every imaginable screenshot included "just in case." The primal urge to document everything seems to be too hard for many to resist. (It is exactly inversely proportional to the urge to resist reading it, however.)

Finally, we have the ultimately unforgivable transgression of bloviation. Why do writers write, "For more in-depth information about this topic, please refer to..." when "See..." will do? Must we really tell users, "To use this product to its fullest potential, you must think about how its abilities apply to your particular organization?" Is there a system administrator alive who feels informed by "The interface is the mechanism through which you view, enter, and interact with the information stored in the database?" I think not.

What's so bad about volume, anyway?

Content volumization has horrible consequences in three main areas that every organization battles constantly: time, cost, and usability. These are interrelated, of course, but even taken individually, they are shocking in their deleteriousness.

Time is in short supply, yet we waste it at every turn in content development. Notwithstanding Pascal's truism, making more content usually takes longer, particularly from a global (not just multilingual) viewpoint. Authoring a piece of content is actually only the beginning. The content must be reviewed (even if only by the author), laid out, processed (bookmarks, cross-references, index markers, tags, etc.), checked in and out, QA'd, finally assembled, etc. Add it all up, and a 5-minute field label rewrite can become a 16-hour company chore, as you can see from the accompanying table (Figure 1 on next page).² Of course, this applies only to one field label in the first version. If there are three new releases per year and 1,000 field labels, well, you do the math.

Now consider localization. Volumized content is processed by localizers (counted, compared, leveraged, etc.), translated, reviewed, laid out again, processed again, checked in and out again, QA'd, finally assembled, etc. These tasks in localization are done per language, so the time to do them multiplies geometrically. Looking at the Time column in Figure 1, you see that a simple 5-minute addition for this project actually takes 1,600 times longer than the authoring effort.

Closely related to wasted time in volumizing content is cost. All the extra time spent on the content is, of course, more expensive. Localization costs are based primarily on volume, so author bloat raises content costs dramatically. The Cost column in Figure 1 shows that 5 dollars worth of content costs 129 times as much to implement fully in the original language, and a whopping 1,089 times as much to implement globally. (Wow!)

Those are just the direct costs; closely related are costs like printing and deployment. What if you could reduce a 40-page install guide to a tri-fold card? (Don't smirk—I've done it.) Imagine the print savings. Then imagine more distantly related costs: tech support calls, marketing communications, knowledgebases, Web content, etc. With single-sourcing and content automation, direct costs may drop by orders of magnitude, but the content volume itself is still the driving cost factor.

Last, and (stupidly) perceived as least, is the effect of volumization on usability. It goes back to the time squeeze. If users must read content (Heaven forbid!), they want the information fast and easy. The higher the volume, the slower and harder it is to find anything.

Jakob Nielsen, the Web usability guru, has performed countless studies to demonstrate one undeniable fact:

Content usability and volume are inversely proportional.³ Moreover, greater volume almost always means more authoring sessions and more authoring sources. This multiplication of inputs raises the chances of human error, inconsistencies, scattered information, and the like, lowering usability even more (not to mention translation memory effectiveness).

What can you do about volume?

No matter what your position in an enterprise (technical writer, manager, VP of product development, or CEO), do not for one moment assume that content volumization is inevitable. Most content costing decisions today are based on the mistaken assumption that volume is a constant. If that's the case, the thinking goes, why not apply the lowest cost factor possible?

Meanwhile, jobs march inevitably offshore. All this does is lower production costs for now (a defensible result only in the short term). The content that results from this effort is at best no better than before, and usually much worse and longer. "Offshoring" may reduce content production costs, but it can actually raise localization costs and time, offsetting some offshoring gains.

With volumization no longer inevitable in your mind, take the right steps to solve the problem you actually have. If you think technology is the solution to your problem, then you either don't have a problem (unlikely), or you don't really know what your problem is. Volume is a process problem; it is a people problem. Technology will not solve the problem, not any more than buying a new refrigerator will make your great aunt's fruitcake disappear. Reducing the volume of content starts with changing authors' conceptions, habits, drivers, and skills. It is not an event, but a lifestyle change.

Start connecting content development and localization more closely. Assemble both teams regularly to discuss problems. Create a feedback loop—in both directions. Exchange metrics (see below). Empower teams to solve real problems, maybe even in the product itself (Wow again!). Cross train people in single sourcing, translation memory (TM) tools, authoring tools, etc. The siloed nature of both content development and localization creates huge obstacles to efficiency and cost-effectiveness.

You can't fix what you can't measure, so measure everything. Then focus on the metrics for the right things: word count, page count, graphics count, screenshots, time on task, staff costs, touch-points. These paint the true picture of content. Then multiply those metrics by your expected multilingual workflow.

A company I know calculated that their production cost for incorporating a single screenshot in a single language was \$15. Their localization requirement currently stands at 19 languages. One manual for one of their products contains 300 screenshots. Again, I leave you to do the

Figure 1

Person	Comments	Instances	Time	Cost
Programmer (\$60/hr)	5 min. rewriting field label	1	0.083 hours	\$5
Writer (\$40/hr)	15 min. research 30 min. writing/editing 15 min. to locate in multiple locations	8	8 hours	\$320
Editor (\$40/hr)	15 min. of disgust 30 min. writing/editing 15 min. to locate in multiple locations	8	8 hours	\$320
Subtotal			16.083 hours	\$645
Localizer (\$40/hr)	15 min. research 30 min. writing/editing 15 min. to locate in multiple locations	8	120 hours	\$320 $x 15 \text{ languages} = \$4,800$
Total			136.083 hours	\$5,445

math. How much do screenshots cost in your organization? If you don't know, you just touched your real problem.

Finally, you must—absolutely must—think globally, not just "multilingually." Connect content development with your other business processes: tech support, product development, marketing, human resources, training, and so on. Where does content originate? Where does it go? How is it tracked from release to release? Calculate the life cycle of products, the number of releases planned, and how the product will change. How does all this affect the metrics, usability, and workflow for content development? Like product development, understanding these links and planning are the keys to less, but more effective content. The worst thing your teams can do is just start writing.

Metrics are critical to global thinking, but failing to measure the right things can lead you astray. Companies usually have exact localization metrics (more irony here) because localizers live and die by metrics, especially TM percentages. Too often, though, organizations fall into the TM trap, letting leveragability drive their content decisions, instead of the other way around. Ignoring usability (I don't recommend it), the value of leveraged TM is just a metric like any other. The decision to rewrite content, reduce volume, and improve usability should include, but never be driven exclusively by, TM percentages. Most organizations, however, do just that,

and it will prevent them from ever getting out of the content volumization spiral. As Abraham Maslow said, "If all you have is a hammer, you tend to see every problem as a nail." TM is not, and should not be, your only tool on the content workbench.

As you can see, content volumization is not inevitable, but it is disastrous and costly. Like male pattern baldness, it's a problem that most people think is less obvious than it actually is. If you want, get some mousse and give your hair a nice, full look, but start cutting that content now.

Hans Fenstermacher is President and founder of ArchiText, a language service provider. He has developed ABREVE, a methodology for reducing volume and globalizing content. Every other month, Hans will share his insights into the link between authoring and localization in CSN magazine. Contact Hans at hansf@architext-usa.com.

¹ All quoted content in this article comes from real, published documentation. The identifying characteristics have been changed to protect the perpetrators.

² The table represents actual calculations made by Ben Martin at JD Edwards prior to implementing a radical enterprise-wide global content development process.

³ Visit <http://www.useit.com/papers/webwriting/writing.html> for a fascinating study on this topic.

tech writing focus

matches?

we **don't** need

no stinkin' matches

Your mother told you never to play with matches, but I'm guessing your mother never authored or translated technical content. In the world of authoring and translation, matches are a lot more practical than the ones she was talking about. Not only should you play with these matches, but in the world of Translation Memory (TM) you could get burned *not* playing with matches.

Matchmaking: a Quick Primer

In TM jargon, "matches" refer to segments of text that are the same or similar to previously identified segments of text. Once a segment is translated, the segment pair (source + target languages) is stored in a database (the TM). Later, when an identical segment appears, the TM is called up, and you can leverage the translation from memory. You've met your match, so to speak.

Matches come in several shapes and sizes. Identical matches (100%) are just that: entirely identical to a previous segment. For example, the segment *To continue, click Next* is a 100% match with *To continue, click Next*.

When segments don't match completely, math enters the picture, but it's fuzzy (in addition to the Internet, Al Gore claims to have invented TM). Non-100% matches are "fuzzy matches." For example, *To continue, click Next* versus *To continue, click the Next button*. The latter segment matches the former at a given percentage (say, 70%), which is calculated differently depending on (a) the TM program you're using, (b) the segmentation rules used, (c) the weather, (d) all of the above.

By Hans Fenstermacher

why content authors should care about TM

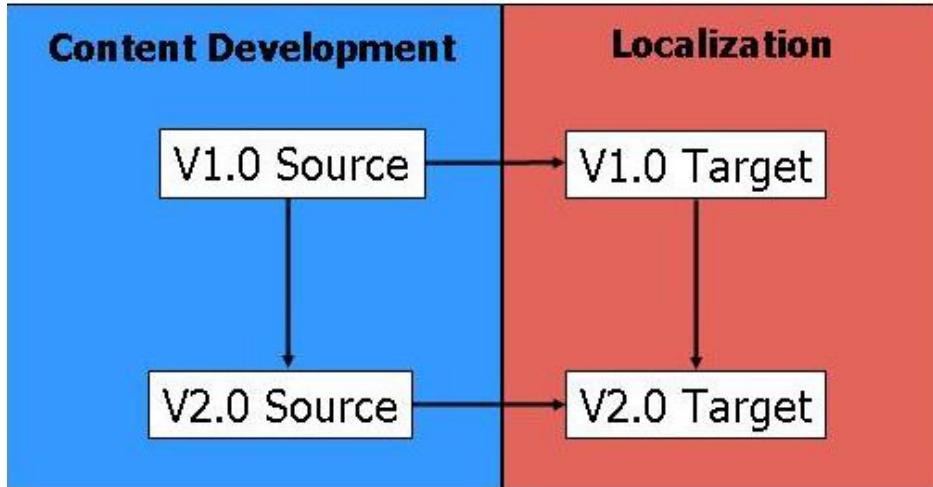
Segments that are 100% matches with other segments in the source language are called "internal repetitions." These are like regular TM-type 100% matches, only they're missing their target-language pairs. But you know you will be able to leverage them after you translate the first instance. Segments that are not 100%, fuzzy, or internal repetitions are "new," meaning they must be translated from scratch.

So, it stands to reason that the less "new" material you have, the less costly (and time-consuming) translation should be. In the localization game, matches increase in cost, starting at 100%/internal reps (least expensive), moving to fuzzy (more expensive—there can be multiple levels of fuzzy), and then on to new matches (most expensive).

You still can get burned

In the ideal world, translation memory, single-sourcing, and 100% matches grow and converge, generally gamboling together in perfect, sparkling harmony. (See figure 1.) It's a beautiful thing.

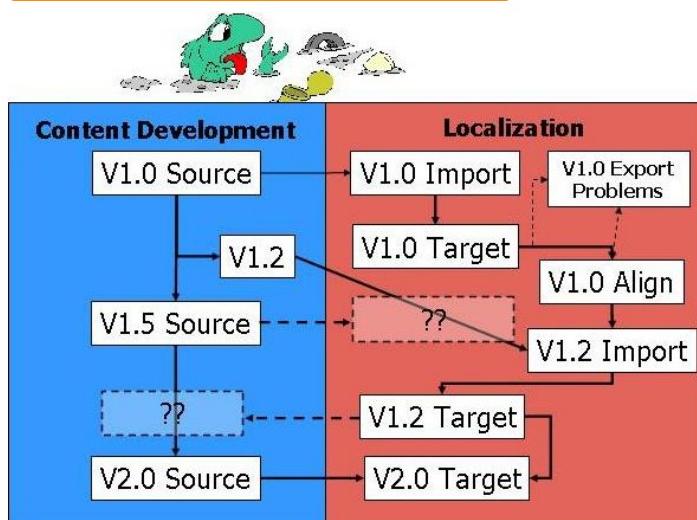
Figure 1: Sparkling perfect harmony.



If you're counting on seeing this rosy picture a lot, however, you're probably delusional.

Why? For starters, there is a failure to communicate. Amazingly, content authors and localizers—so-called specialists in communication—do very little actual communicating. When authors make changes in the source language without corresponding updates on the target side, the content becomes misaligned. This could happen, for instance, if authors create an interim version (say V1.2) that does not get localized, then decide to localize V1.5 some time later. The V1.5 update is "minor"

Figure 2: Distinctly messy place.



and their expectation of localization cost and time is "minor," too. But the localizers never heard of V1.2, so they are comparing V1.5 with V1.0. Both sides are in for a rude awakening. The real world of content development is ultimately a distinctly messy place, as in figure 2.

It gets worse. Often, authors don't understand what TM is and how it works. Translation Memory is a tool, and not

a particularly sophisticated one. It isn't automated like Machine Translation (MT). It isn't artificially intelligent (like certain parts of the Pentagon). In fact, it doesn't do anything on its own. TM doesn't change translation quality per se, it just keeps the quality consistent, bad or good.

When you change translation vendors because of translation quality problems, and then expect your new vendor to charge less because you provided the TM from the previous vendor, you have missed the point entirely. Additionally, TM does not make up for shortcomings in the source content. Bad writing stays

bad, and inconsistencies stay inconsistent, except that they'll be more consistently inconsistent, if you know what I mean.

Finally, TM falls victim—like everything else in our society—to over-reliance on technology. This tool is dumb. For instance, TM (for that matter, MT) cannot resolve ambiguities: the segment *I can fish* (meaning *I can catch fish*) is a 100% match with *I can fish* (meaning *I put fish in cans*), but they obviously are not the same thing.

Additionally, fuzzy matches only reveal similarities in terms of keystrokes: the segment *Check whether the file is corrupt and unusable* is 100% different from *The file is corrupt and unusable*, but TM will see a 75% match. And TM has no way of telling that *From the File menu, select Print* is 100% identical in meaning to *Choose the print option in the File menu*, which is well below 50% match, and therefore considered new.

Prevention

Why should content authors care about all this? There are many reasons, but here are the two most important: money and usability. As we have established, 100%

What's a globally minded author to do then? There are a few tips for moving your content much closer to localization nirvana on the next page...

matches are the least expensive translation option. So, the ideal multilingual content process, like the classic perfect baseball game, is one in which nothing happens. That is, there is no new text (written or translated); every bit of content comes from somewhere else. The "perfect" author, by extension, spends all his/her time assembling content segments, not writing. (Hold on a minute while I increase my disk space for all the angry e-mail I'm about to get...) That is, after all, the ultimate goal of single-sourcing. Want to demonstrate some real value? Don't write. Reuse.

TM is another place where money and usability are perfectly matched (can't... stop... the puns...). Translation memory thrives on consistency—as do users! Creating reusable 100% matches like *Insert the CD*, *Enter these values*; and *Contact your sales representative* is financially sound, and they offer consistent cues to users, who are grateful not to have to struggle with rampant synonyms or stylistic hobgoblins worthy of Emerson.

A match made in heaven

What's a globally minded author to do then? Here are a few tips for moving your content much closer to localization nirvana:

Enough with synonyms - Broaden your definition of terminology to include phrases, and don't deviate from them. The following are unnecessary fuzzy matches: *log in* vs. *log on*; *online mode* vs. *online state*; *click the Date/Time button* vs. *lick the Date/Time button* (it may be just a typo, but fuzzy is fuzzy, folks).

Quit fooling with formatting - Reusing content, and then changing its formatting can dramatically lower your match hit-rate (like putting a pair of fuzzy dice in a Ferrari). The following are *not* 100% matches: *Click the Next button* vs. *Click the Next button*; *Press the <CR><LF> right arrow* vs. *Press the right arrow*; the *Print* option vs. *the print option*.

Really single-source - Create segments that can be used in many contexts, not just one. The segment *Do not let wires block panels* is much less reusable than *Do not block panels*. Introduce screenshots with *The following window appears*: and you'll stop caring once and for all how many times the developers change the window titles.

Genericize - Learn to prefer less specificity when it won't create confusion. Change *Use the "Print" option* to *Use the print option* (and neither you nor the localizers will have to look up "Print" in the UI again). There's no need to state the number in *The three types of networks are*.... If the number of types changes in the next release—and you can bet it will—this is one edit you won't have to make.

Come to think of it, your mother was right: She wanted to keep you from getting burned. But go ahead and play with TM matches; tone down those verbal pyrotechnics. The next time a localizer asks, "Got a match?" You can say with confidence, "Yeah, my content and yours."

Hans is founder and president of ArchiTExt Inc. ArchiTExt specializes in localization, globalization, and end-to-end global content solutions based on ABREVE®.

For more information on ArchiTExt or ABREVE®,
please contact:
Chad LaCroix, Sales Coordinator
Tel. +1 978.409.6112 or toll-free 1.800.688.7205
chad@architext-usa.com
<http://www.architext-usa.com>

food *for* thought

By Hans Fenstermacher



When you think of a "culture," you usually picture ethnic, national, religious, or geographic characteristics that are typical of a particular group of people. Usually, a culture is also distinguished by language, so it's natural in the context of localization to view culture problems as linguistic ones. One way to globalize technical content, therefore, is to remove things that specifically refer to a culture (mom, baseball, apple pie, that sort of thing).

Often, that's pretty easy to do (apple pie doesn't tend to figure prominently in technical writing). Trouble is, this traditional concept of culture only partially addresses deep globalization problems that can bedevil content. To identify and overcome these problems, you need to have a much broader understanding of what culture is.

What is culture?

In technical content, culture should be defined as anything that creates unnecessary bias in favor of a particular user group. Let's take this definition apart to understand all its implications. First, culture is bias, slanting the text in a contextual direction that only its intended audience understands. Bias works like a secret handshake between the writer and specific readers:

"The right use of type can make the difference between your message being understood and it being bounced off the cubicle wall for two points into the trash." (From online help for a font management utility.)

Leave Culture in Your Yogurt and Out of Your Content!

This sentence speaks squarely to male, windowless-office workers who are basketball fans. Everybody else is out of luck.

The second key element in the definition of culture is that the bias is unnecessary. Let's face it, sometimes the right kind of bias is critical to successful content. A user guide for a sophisticated statistical software package can darn well assume that its readers know a thing or two about statistics. But slanting content unnecessarily is culturally inappropriate and simply detracts. (See example below)



NOTE: It is important for you to go through each option in the International Preferences window, in order. Why? Good question. Well, you want to be sure that you make all changes necessary. For example, if you change the language to Spanish, you do not want the [Units] option to remain US English. In Spain, they use the metric system! Everything needs to match.

This presumptive chumminess (personally, with this talking cracker, I think we've entered a weird area here) is completely gratuitous, adding cultural overtones that undermine the seriousness and clarity of the material. It will never work in other languages.

Finally, the broader definition of culture in technical content singles out a particular user group for the bias. For example, it may appeal (with a wink and a nod) to national or linguistic affinity:

"The 5 Whys and Mile-deep Thinking"
(From a 6-Sigma training guide.)

Residents in the current or former British Empire have no trouble fathoming this. But what about, say, Germans? Intellectually, the concept of a mile is probably well-enough understood, but viscerally it's a non-starter. The literally minded will concoct something like *Fünfmal warum und 1,6-Kilometer-Denken*, but that little absurdity evokes little more than laughter. These concepts would need total cultural adaptation in each target language in order to make any sense. Rephrase them, though, and the problem goes away: "Understanding the Causes of the Problem" (as a suggestion).

When culture problems occur in technical content, they take many forms, some more obvious than others. Universally, however, they feel awkward in the source language, and in localization, they invariably become pronounced and problematic in terms of the mechanics of translation or usability, or both. Here are a few types:

Over-specification

Precision is the hallmark of technical writing, so what could be more precise than being as specific as possible?

Being less specific, for one. There are times when slightly less precision actually makes things easier to understand and avoids a culture problem:

"Insert the unit in the opening.
Leave about 3/16 of an inch of
space between the back of the
unit and the wall." (From a
hardware installation guide.)

How much is "about 3/16 of an inch?" How about 2/16? Half an inch? How would you measure that anyway? The problem gets worse when you localize. What happens to this in metric units? *Leave about 4.8 mm of space... is ridiculous.*

Hans will be attending the 2005 CSN Expo in New Orleans, Louisiana, March 31–April 1. If you would like to meet with Hans during the CSN Expo, please contact Chad LaCroix at chad@architext-usa.com to schedule a meeting.

Should you round to 5 mm? Now we're back to, "How much is about 5mm?" and "How would you measure that?" What the writer meant, of course, was, "Make sure the unit does not touch the wall." Well, why didn't she just say that?

Patronization

Closely related to over-specification is information that ends up patronizing the user. Take the case of a specific task aimed at a clearly defined user group. For example, a UNIX system installation aimed at UNIX system administrators:

"Step 1: Insert the CD into the CD-ROM drive tray, label side up."
(From a UNIX system administrator install guide.)

Unless the target readership is under the mental age of eight, this over-specification is insulting. Not only that, it uses more words to make the point, which adds work and cost for localization.

Inappropriate information

A culture problem often develops when content includes information that simply doesn't belong. This can happen in many ways, but usually either when the user group is poorly defined at the outset (blame the writers), or information is "forced" into the content by other stakeholders (blame the product developers). Here is an example of content that suddenly veers away from the expected knowledgebase of the targeted user group (the average salesperson) and crosses well into IT technical territory:

"Consult your PC documentation to determine your parallel port's mode and change it to ECP, or change modes in the BIOS:

1. Access the BIOS when the system is booting by pressing a key or key combinations (commonly Esc, F1, F2 or Del).
2. In the BIOS, change the setting for the parallel port mode." (From a scanner user guide.)

I don't know about anyone else out there, but as an experienced computer user, I am extremely reluctant to go fiddling with the BIOS. Blithely including these instructions in user documentation looks like an invitation to disaster. (And you have to wonder how many tech support calls this instruction has generated from confused sales reps who suddenly can't boot their PCs anymore.)

Jargon

Let's be clear on this culture problem: Terminology is one thing, jargon another. The former is a set of agreed-upon terms used to describe specialized elements in an industry or technical area. The latter is the informal, casual "stuff" experts say to each other when they both know exactly what they mean. Jargon nearly always can and should be eschewed for its terminological equivalent. Take these examples:

Jargon	Terminological Equivalent
• <i>in the ballpark</i>	• <i>approximate</i>
• <i>best of breed</i>	• <i>high quality</i>
• <i>horsepower</i>	• <i>capacity / output</i>
• <i>cooperates at the software level</i>	• [Darned if I know!]

There is a place for jargon, but it isn't in technical content. There is something to be said for its use in marketing collateral (where precision often takes a backseat to schmoozing with the audience). Most of the time, technical content is aimed at a much broader audience than just experts, and even if it isn't, jargon is a son-of-a-gun to localize (see what I mean?).

Creating uncultured content—I mean that in a good way!—isn't easy, but taking a critical look at how we go about it will help us all improve customer experience and smooth out the multilingual content process. It has been said that "culture is what remains when we forget everything else." We would be wise to forget culture (leaving it in our yogurt where it belongs) and to focus on remembering "everything else," which is hard enough as it is.

Hans founded ArchiTExt® in 1994 after a successful 15-year career in technical communication and translation, including experience with some of today's industry leaders.

He started a different kind of company because he realized that localizers were really in the same business as their clients: building good communications with customers all over the world. Since then, ArchiTExt has grown into a well-established provider of localization and multilingual services to Fortune 2000 companies.

Content Optimization

why technology can't do the job

The technical authoring and localization industry seems to focus on technology as the primary solution to most problems. In a way, who can blame us? After all, the high-tech sector generally has been the leader in innovation. Our industry tools have significantly boosted productivity and efficiency. Now that content optimization looms large as the next big problem to tackle, many of us are naturally depending on technology to solve this problem, too.

Unfortunately, we're headed for frustration. Technology cannot lead the content optimization mission, because the biggest problems in content aren't technical. When it comes to content, technology has fundamental shortcomings. The sooner we realize this truth, the better we will be able to solve the challenges before us in real ways.

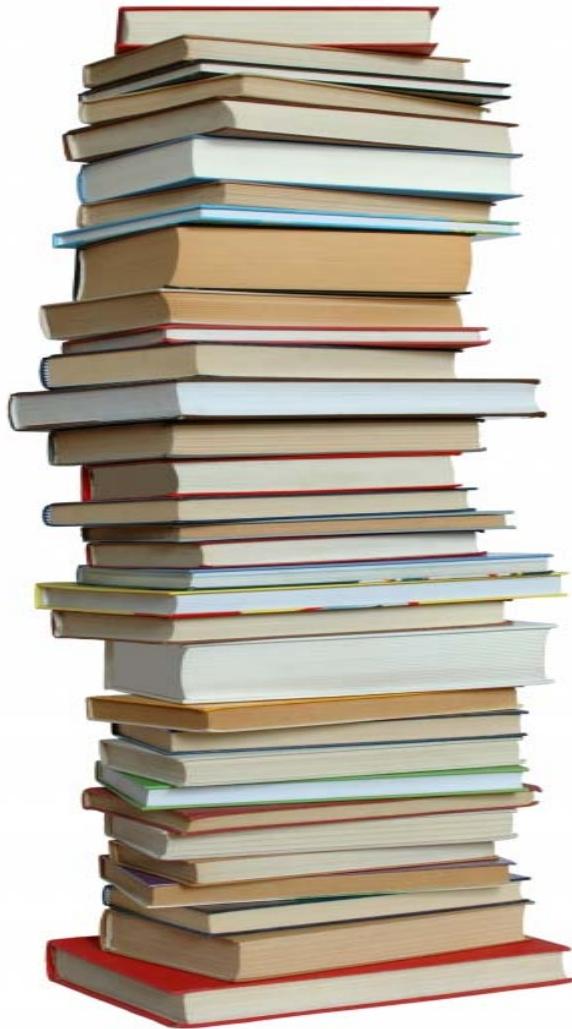
Let's look at some of those shortcomings.

Technology can't resolve ambiguity.

Natural language, the primary medium for all content, is full of ambiguity. Technical authors do everything they can to avoid it, but they don't always succeed. Take this sentence: "The label is on either side of the drum." This could mean that there are two labels, one on each side of the drum, or it could mean there is only one label, which may be on one side of the drum or the other. Without more information, either interpretation is correct, but both cannot be. (Side note: This example comes from real documentation. When asked which interpretation was correct, the author responded, "I don't know!").

There is no technology or tool that can resolve this kind of ambiguity, because human beings can't resolve it either. The sentence is grammatically correct—it may even be terminologically correct—but it isn't unequivocal. For translators, this means they must choose one of the two interpretations. Given this 50-50 probability, it's likely that half the translations will be wrong.

By Hans Fenstermacher



Technology doesn't understand anything.

Another thing technology can't do is actually think. Tools often give the impression that they are thinking, but they are merely comparing things (occurrences, odds, rules, etc.). Consider the following sentence: "The main purpose of the payroll run is to calculate employee pay correctly at regular intervals." From this sentence, a content-analysis tool might "understand" that payroll run is a process, that this process involves an action (calculate employee pay), and that this action occurs with some frequency (at regular intervals). Fine. But what the tool doesn't know is that this sentence is completely uninformative to actual human beings; anyone who's ever held a regular-paying job knows what the purpose of a payroll run is.

By not understanding what it is processing, technology fails to tackle one of the most fundamental problems in content optimization: usability. Technology can make content consistent, more uniform, more structured, and (technically, perhaps) error-free, but it cannot make content informative. Only a human editor knows what this sentence really needs—to cease to exist.

Technology doesn't care about usability.

Technology doesn't even know what usability is. Tools that manage content structure and workflow are designed to follow rules (ironically, rules that must be created by human beings), the outcomes of which are binary—yes/no, allowed/prohibited, accepted/rejected, 1/0. But usability isn't binary; it's full of complex issues like context, exceptions, interpretations, and so on. The creators of content (humans) are not binary, and the recipients of content (humans) are not binary. Why would we expect the binary technology in between to ensure usability?

Usability requires understanding who the recipients of information are and what they need. Consider again the payroll run sentence: all the binary rules may be followed precisely for formulating this information, but only the creators and recipients know that the information is useless (usability = 0). No technology is actually intelligent enough to delete this sentence.

Technology can't restructure content.

Content structure is a key to its usability. One of the critical tasks technical authors perform is analyzing and organizing information to make it usable. Technology simply can't do that. Take this example:

"To embed a graphic in a text document, you must open the object in its native application, remove all font information (for example, converting fonts to outlines), and save the file in JPEG format. Then in the text document, place your cursor where you want the graphic, and use Import to insert the JPEG file you just created."

This highly unreadable (dare I say off-putting?) text block becomes usable with good structure:

To embed a graphic in a text document:

1. Open the object in its native application.
2. Remove all font information (for example, convert fonts to outlines).
3. Save the file in JPEG format.
4. Open the text document.
5. Place your cursor where you want the graphic.
6. Use Import to insert the JPEG file you just created.

Technical authors apply these techniques almost instinctively, but tools don't have a clue.

Technology can't self-correct.

Tools are very precise; that is their strength. Tools also rely on precision implicitly; that is their weakness. All content processed by tools originates from a human being, and human beings are by nature imprecise. This imprecision frequently exceeds technology's tolerances, but it's far less of a problem for other humans. How many animals of each type did Moses take on the ark with him? The correct answer is none, because it was Noah who built the ark, not Moses. Many of you, though, undoubtedly self-corrected that error without even knowing it.

Hans developed ABREVE® in response to an expressed need for new ways to reduce localization costs while preserving the integrity of content. ABREVE® addresses these issues directly while improving the content's usability.

ABREVE® is a patent-pending methodology that enables low-risk step changes that can provide drastic localization cost reductions.

“Technology is only one of the tools for content optimization. Remember Abraham Maslow’s dictum: If all you have is a hammer, you tend to see every problem as a nail.”

Authors of technical content are trained to be as precise as possible, but nobody's perfect. Minor errors are quite commonplace, as these examples show (corrections provided in parentheses):

- Then lick the Date/Time Stamp button. (click)
- The air damn directs the airflow. (air dam)
- You must not turn off the printer. (now)

Relying on spell-checkers (which also demand complete precision) cannot save the above or countless other examples of misinformation. Incidentally, after localization, this information can no longer be self-corrected by readers, because the cause of the imprecision is literally lost in translation.

Technology can't reduce volume.

Strictly speaking, tools can limit volume by imposing constraints. Proponents of text analysis and processing tools say that volume limits demonstrate the value of the technology (you know, maximum 7 words per sentence, that kind of thing). But artificial limits do little to optimize content. The philosopher Bertrand Russell famously showed that a few words can be just as uninformative as many:

Colorless, green ideas sleep furiously.

Real volume reduction requires a deep understanding of content: How can text be restructured, combined, and rearranged to make it shorter? What information is unnecessary because it is superfluous or redundant (remember: redundancy is a hallmark of technology)? Only human beings can answer these questions. If fact, users (also human beings) urge us to do so every day by refusing to read the documentation we produce.

Technology isn't natural.

Last, but not least, let us not underestimate human nature. Out of the last 2 million years of our evolution,

we have spent—statistically speaking—approximately 0% of that time in the digital age. So, every technology is designed to mimic the analog world, because if something doesn't look and feel "natural," we won't use it. Writing is a deeply analog task, and humans fiercely resist writing techniques that aren't natural. That's why controlled language hasn't gained much in popularity among technical authors in over 30 years; it just runs too deeply against every natural instinct writers have. For technology to truly succeed, it must feel natural, so we will want to use it.

Has your car ever driven you to work? Has your washer ever done the laundry for you? Have hardware and software ever gotten together and written a user manual while you weren't looking? Of course not. They need you to do those things.

Technology is only one of the tools for content optimization. Remember Abraham Maslow's dictum: If all you have is a hammer, you tend to see every problem as a nail.

The problem of content optimization is complex, and the problem requires complex, not pie-in-the-sky, solutions. Content management and single-sourcing proponents often hold the view that technology will solve the content optimization problem. They seem to think that optimizing the workflow around content is the same thing as optimizing the content itself. Anyone who has ever read a user manual, however, knows that just isn't true.

Born and raised in Germany, Hans is a graduate of Princeton University and the Fletcher School of Law and Diplomacy. He speaks six languages and has been in the language business for over 25 years.

A Communication Revolution Is Under Way.

High-performance communication drives successful global organizations.

And communication is made up of messages crafted by people – not by technologies, systems, or workflows.

Imagine your messages going from you to your customers better and faster. Now imagine a process that gives you all that and saves your company money – substantial, revolutionary savings you may only dream about.

Welcome to ABREVE®.

The power to improve your global content, empower you, and thrill your CEO.

ABREVE®

Communicating.

Better.

ABREVE®

The Next “Big Thing” in Localization

More and more client-side localization managers agree: What we need next is not more technology, but better processes and, above all, better content.

The patented ABREVE® process from ArchiTText addresses exactly this issue by targeting three elements: increased content quality (usability), lower content volume, and optimal localization readiness.

Technology alone does not, and will not, achieve those goals.

Content usability depends on factors like clarity, consistency, accuracy, ease of use, and more. But technology cannot determine whether information is easy to understand, consistent, or simply wrong or unnecessary.

ABREVE® services permit development of content that better meets global users' information needs by combining best practices in content development with effective workflows, localization expertise, proprietary processes, and technology.

Content volume has a huge effect on usability. If users cannot find information because there are too many words, accuracy and ease of use are irrelevant. Word count is also the single biggest factor in localization costs.

ABREVE® targets word-count reduction through a series of struc-

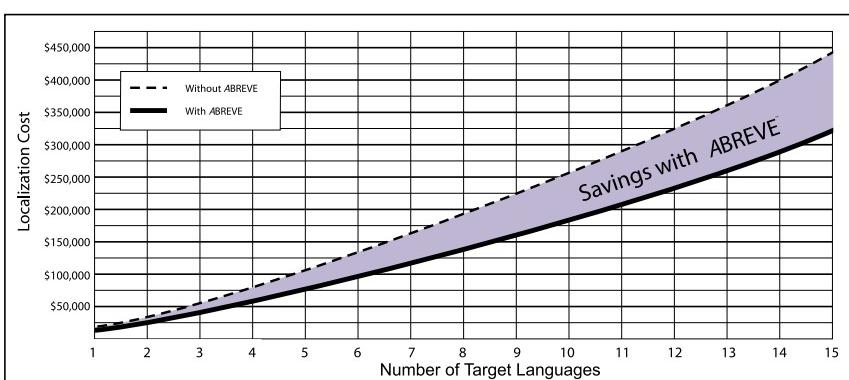
tured steps, eliminating redundancies, simplifying language, and ending the domination of format over content.

Word-count reductions of 25–60% through ABREVE® services are routine, and those reductions translate directly into an improved return on investment in localization (see graph).

Today's content development approaches are not generally designed to meet the challenges of localization. ABREVE® focuses on making content ready for its path through localization by removing obstacles (like inconsistent terminology), optimizing tool use (through more 100% matches), and anticipating downstream work (with optimized graphics and formatting).

Because ABREVE® is not tied to a particular tool, it can improve any technology-based workflow, whether it includes CMS, TM, or MT. And because the ABREVE® process targets the inputs into the content deployment process – rather than outputs – it can achieve equally dramatic results with in-house, contract, or offshore content development strategies.

For information on ABREVE®-based services, case studies, metrics, and testimonials, contact ArchiTText at abreve@architext-usa.com or toll-free at 1.800.688.7205.



Create. Integrate. Communicate.

ArchiText®
www.architext-usa.com

there's no info

By Hans Fenstermacher

like no-info

Content authors are in the information business. Their jobs are to create, architect, design, write, localize, deliver, and ultimately consume information. They relentlessly meet the voracious demands of the Information Age, but they also have become purveyors of a kind of information twaddle—call it *No-information* (why beat around the bush?)—that threatens the quality of their work.

To understand *No-information*, let's first be clear on what information is. Like obscenity, information is hard to define, but (also like obscenity) we tend to recognize it when we see it. Simply put, information is that which tells us something we didn't already know.

No-information, on the other hand, looks, feels, and sounds a lot like information. No-information requires the same amount of time, resources, and development as information. No-information additionally takes the same amount of resources, time, and money to localize. And end-users will likely give No-information the same amount of attention and time as real information.

Come to think of it, No-information is exactly like regular information in every respect, except one: it doesn't inform you of anything (witness the very first sentence in this article!) No-information is treacherous because it takes away time, resources, money, and attention from everyone who touches it. It's also exasperating to end-users, who instinctively recognize it for what it is not: informative. Here are some of the many forms No-information can take (all examples are from real content):

Obvious No-information

The simplest form of No-information is the obvious form. This means facts, data, or concepts that are well known by or self evident to the intended recipients. Obvious No-information comes in all shapes and sizes:

From an enterprise software setup guide: "*This guide is designed to assist you in setting up XYZ. There are many ways to set up and use XYZ. At [our company], we use XYZ to run our software business, which is in some ways similar to many businesses and in some ways very different.*"

Notice the company's example of how it uses its own software? Even that excludes any real information!

From a software user guide: "*Using computers within the corporate environment has made it easy to create documents that might contain mission-critical information.*"

It's difficult to argue with that; it's equally difficult to learn anything from it.

From a UNIX system administrator's guide: "Place the CD in the CD-ROM drive label side up."

Surely, you jest!

Tautological No-information

No-information can also be camouflaged by truth and accuracy. Much to our great chagrin, however, truth and accuracy are not sufficient or even necessary conditions for information. All information is required to do is tell us something we don't know. On this score, No-information tautologies (redundant repetitions) give real information the widest possible berth. To wit:

From an operator's guide: "*Using other settings may cause very random and unpredictable results.*"

Except possibly for the subtlest of distinctions among persnickety statisticians, "random and unpredictable" is a redundant repetition. Furthermore, the qualification of random with "very" is gilding the lily, to say the least.

From a modem user guide: "*This modem provides reliable data communication to meet a variety of needs.*"

By definition, modems provide data communication, presumably doing so reliably. Sadly, the exact nature of the "variety" of needs this modem meets is left to our overactive imaginations.

No-information—once again with feeling

The expansive repertoire of No-information includes another form similar to the first two above. Here, previously known information, or even previously known No-information, is freshly restated as if this new presentation would somehow magically infuse the No-information with meaning. Generally, this phenomenon is attributed to a lack of editing (a get-out-of-jail-free card that technical writers produce far too readily and far too often,) but it may actually originate in the writer's primal urge to find the real meaning that they know deep-down isn't there.

Header from a hardware guide:
"Troubleshooting Problems"

Does the product unusually require troubleshooting those things that work properly?

From any software user guide: "Close any open software programs that are running on your computer."

The writer felt the need to explain that "closing" is something you do to things that are open, and that open programs are those that are running on a computer. Presumably, "Close all programs" would have been just too short and obviously ill-mannered.

S&M-style No-information

In today's competitive marketplace, there is an imperative to constantly reinforce product advantage and superiority. Here, No-information digs in its stiletto S&M heels (umm..., that's Sales & Marketing...).

Most user content is created to accompany products that have already been sold. Why, then, do content authors insist on reinforcing the wisdom of the user's purchasing decision? This form of No-information comes in small occurrences, through editorializing adjectives or judgmental interjections such as *Welcome!* or *Thank you!* Or *Congratulations!* or even in lengthy paeans to the product. Either way, S&M No-information adds cost and time to every phase of content development, and then robs the end user of his or her anticipated reward (information). Note this example from a printing system guide: "*Welcome! The XYZ User Guide was created to provide you with information and demonstrate the key features of the XYZ application. XYZ is a robust, state-of-the-art system for... [bulleted list]*"

Does anyone ever read that stuff?

Hans is also founding
Chairman
of the Board of the
Globalization and
Localization Association
(GALA).

Founded in 2002, GALA
is a fully representative,
non-profit, international
industry association for
the translation,
internationalization,
localization, and
globalization community.

"Templates should exist to provide a framework for content authors, not a straightjacket.

Templates should be tight enough to keep information from falling all over the place, but loose enough to allow common sense to fit in. "

Bridge over the River No-information

No-information also stems from the urges in all content authors for logic and structure. One of the most common is the "bridging" urge, the desire to smooth over perceived abrupt shifts in a linear thought process. This urge is admirable—provided the writing is being graded by a teacher on the use of the "topic sentence." But in the real world of technical content, linearity is often unnecessary, impossible, or even disruptive. End users do not live in a world of narrative prose that must make sense from cover to cover; theirs is a jumble of context-sensitive help, reference material, and troubleshooting—where content is expected to be self-contained and usable in chunks.

Technical authors demonstrate their bridging urge through self-imposed segues that link topics, through introductions designed to avoid mental jack-rabbit starts, and through summaries softly placed to prevent user whiplash when a topic ends. This bridging No-information fails to understand that users do not read sequentially and that other content devices already play whatever minimal bridging role may be needed. For instance, bold headers in a different typeface that visibly introduce the next topic. Here are some classic bridges:

From a software user guide: "*The next section describes how to manage files.*" (*Followed by the subhead "Managing Files."*)

This topic-ending sentence is usually more useful to the writer as a mental sticky note about what's next; too bad it can't be removed as easily.

From an anti-virus software guide:

"Creating a Schedule to Update XYZ

"Creating a schedule to update XYZ is a simple procedure: ..." (followed by two steps.)

Two steps qualify as "simple" in anyone's book. Why it's necessary to introduce and characterize the two steps is beyond me.

No-information on a (tem)plate

The last No-information type, like the previous one, has a single-minded purpose: impose structure at all costs. This goal is usually handily accomplished (including the cost part) by allowing a predefined template to control content decisions. Overly rigid templates can turn real information into No-information. For instance, take the rule of following every header with a stem sentence (an introduction.) Many times this results in saying the same thing again. Then there's the rule that every table must be formally announced by a descriptive sentence (any self-respecting table title ought to be able to perform this function on its own.) Or the rule requiring that every possible screenshot in a procedure be displayed, whether there's anything to do or see in it or not (ugh.)

Templates should exist to provide a framework for content authors, not a straightjacket. Templates should be tight enough to keep information from falling all over the place, but loose enough to allow common sense to fit in. And when I say common sense, I mean the end user's common sense, not the writer's (can you imagine any "normal" user complaining about a missing stem sentence?!) Here, for your delectation, is an example of No-information served on a template:

Chapter 1 Installing the XYZ

At a Glance

Purpose

This chapter describes how to install and set up the XYZ. It also lists the system requirements for installation and setup.

In this Chapter

This chapter contains the following topics:

- System requirements
- Installing the XYZ
- Setting up the XYZ

Naturally, the guide also contains a complete table of contents and index. It's also noteworthy that the bulleted list for "In this Chapter" does not consist of cross-references, but re-keyed entries. (Go figure.)

Conclusion

No doubt, there are other types of No-information. While they may seem harmless, even amusing, do not be fooled: No-information is a threat to the usability and cost of all technical content. Authors should take pains to eliminate it ruthlessly. The rewards will be great for themselves and, more importantly, for the rest of us.